LogitDynamics: Reliable ViT Error Detection from Layerwise Logit Trajectories

Ido Beigelman
Technion - Israel Institute of Technology, Israel
idobeigelman@campus.technion.ac.il

Moti Freiman
Technion - Israel Institute of Technology, Israel
moti.freiman@bm.technion.ac.il

Abstract

Reliable confidence estimation is critical when deploying vision models. We study error prediction: determining whether an image classifier's output is correct using only signals from a single forward pass. Motivated by internal-signal hallucination detection in large language models, we investigate whether similar depth-wise signals exist in Vision Transformers (ViTs). We propose a simple method that models how class evidence evolves across layers. By attaching lightweight linear heads to intermediate layers, we extract features from the last L layers that capture both the logits of the predicted class and its top-K competitors, as well as statistics describing instability of top-ranked classes across depth. A linear probe trained on these features predicts the error indicator. Across datasets, our method improves or matches AUCPR over baselines and shows stronger cross-dataset generalization while requiring minimal additional computation.

1. Introduction

Reliable confidence estimation is crucial when vision models are deployed in high-stakes settings (e.g., clinical decision support). Despite substantial progress in uncertainty and confidence estimation, modern models can remain overconfident even when wrong, and this behavior often becomes more pronounced under distribution shift.

In this work, we study *error prediction* in a Vision Transfomer (ViT) based image classifier: given an input image x with label y, a trained classifier f outputs $\hat{y} = f(x)$. Our goal is to predict the binary error indicator

$$e(x) = \mathbb{1}[\hat{y} \neq y] \tag{1}$$

using only *internal signals* available from a single forward pass, without external verification at inference time.

Motivated by recent progress on internal-signal hallucination detection in large language models (LLMs), we ask whether analogous depth-wise signals exist in Vision Transformers. To this end, we propose a layer-dynamics method that captures how class evidence and competition evolve across depth, and uses these trajectories to predict when the model's final prediction is likely incorrect.

2. Background

Error prediction and confidence estimation in neural networks. Estimating the reliability of deep neural network predictions has been widely studied in computer vision. Bayesian uncertainty estimation methods approximate the predictive posterior, including Monte Carlo (MC) dropout [9] and deep ensembles [14]. Although effective, these approaches often incur additional computational and memory overhead at inference time. A complementary line of research investigates post-hoc confidence estimation using internal model signals obtained from a single forward pass. Classical approaches rely on logit and softmax-based confidence measures, including maximum logit/softmax, margin-based scores, and energy-based formulations [8, 10, 11, 16, 17]. Other methods operate in the feature space of deep networks, such as the Mahalanobis distance-based scoring originally proposed for out-of-distribution (OOD) detection [15]. Beyond these techniques, several works propose training auxiliary predictors on internal representations or logits to predict misclassification [1, 5].

Hallucination detection in LLMs via internal signals. The deployment of large language models has highlighted hallucinations, where models produce confident but incorrect outputs. Recent work proposes detecting hallucinations using only internal signals (e.g., token probabilities, entropy-based measures, and internal activations), without relying on external verification [2–4, 18].

Motivation for cross-domain transfer. Modern vision systems and LLMs often share transformer-based backbones, suggesting that error-related internal signals (e.g., unstable belief formation across depth) may be measurable in both modalities. This motivates our evaluation of whether internal-signal hallucination detectors transfer to ViT error prediction.

3. Methodology

Classical confidence measures typically rely on the logits of the classifier. While effective, such approaches ignore how class evidence evolves throughout the network. Prior work has shown that intermediate predictions can change across depth and may exhibit "overthinking" behavior [12]. Building on this, we propose **LogitDynamics**, a lightweight error predictor trained on depth-wise logit trajectories and top-K stability features, to model *layer-wise class competition dynamics* in ViTs and better predict when the model is likely to make an incorrect prediction. LogitDynamics extracts logit-based signals across layers and introduces features that capture the model's evolving class beliefs.

3.1. Layer-wise Class Projections

Consider a pretrained ViT with T transformer blocks. Let $h_t(x)$ denote the hidden representation (CLS token) in layer t. To expose intermediate class evidence, we attach a lightweight linear classification head to each layer. Concretely, each head maps $h_t(x)$ to class logits

$$z_t^{\text{head}}(x) \in \mathbb{R}^C,$$

where C is the number of classes. These auxiliary heads are trained to predict the ground-truth label while keeping the backbone frozen.

This procedure yields a sequence of layer-wise class-score vectors $\{z_t^{\text{head}}(x)\}_{t=1}^{T}$ that approximate the model's evolving class beliefs. Let $z^{\text{clf}}(x) \in \mathbb{R}^C$ denote the base model's final classifier logits and define the final predicted class as $\hat{y} = \arg\max_c z_c^{\text{clf}}(x)$. From each of the last L layers, we extract a $(K{+}1)$-dimensional vector consisting of (i) the logit $z_t^{\text{head}}(x)_{\hat{y}}$ and (ii) the top-K logits among classes excluding $\hat{y}$. In addition, we extract the same $(K+1)$-dimensional vector from the final classifier logits $z^{\text{clf}}(x)$. We concatenate these $(L+1)$ vectors to form the primary logit-based features.

3.2. Top-K Dynamics Features

Beyond raw logits, we hypothesize that *instability of the model's top hypotheses across depth* is predictive of errors. We focus on the last L transformer blocks and additionally include the final classifier logits as the terminal element in the sequence. Let $\{\tilde{z}_\ell(x)\}_{\ell=1}^{L+1}$ denote these logit vectors, where $\ell = 1, \ldots, L$ correspond to the last L layer heads and $\ell = L+1$ corresponds to the base classifier.

For each position ℓ, define the top-1 identity c_ℓ (the highest-scoring class) and the top-K set S_ℓ (the K highest-scoring classes). Using these quantities, we compute a small set of features that summarize how the model's leading hypotheses evolve across depth:

- **Top-1 Switch Rate:** how frequently the identity of the top-1 class changes from one depth to the next.
- **Top-K Weighted Jaccard Similarity:** how consistent the *set* of top-K classes remains across depth, weighting classes more when they carry more of the top-K probability mass.
- **Unique Top-K Count:** how many distinct classes ever enter the top-K across the depth sequence, capturing the breadth of competing hypotheses.
- **Top-1 Mode Frequency:** how often the most common top-1 class appears across depth, measuring how strongly the model repeatedly favors a single class.
- **Top-1 Entropy:** how dispersed the top-1 identities are across depth (low entropy indicates consistent top-1 predictions; high entropy indicates volatility).
- **Top-1 Unique Count:** how many distinct classes appear as top-1 at any depth.
- **Top-1 Commitment Depth:** how early the model "locks in" to its final top-1 class and maintains it for the remaining layers (earlier commitment suggests more stable decisions).

Intuitively, correct predictions tend to exhibit a stable top-ranked structure and earlier commitment, whereas errors often involve volatile competition among high-scoring classes. Formal definitions are provided in Appendix A.2.

3.3. Error Prediction Model

We form the final feature vector by concatenating:

1. the concatenation of the $(K+1)$-dimensional logit vectors extracted from each of the last L layer heads together with the corresponding vector from the final classifier logits, yielding $(L+1)(K+1)$ numeric features
2. the proposed competitor-dynamics features.

A linear classifier is trained on these features to predict the binary error indicator of the base model. Importantly, the backbone network remains frozen, and the method requires only a single forward pass at inference time.

Overall, the proposed approach preserves the efficiency of post-hoc confidence estimation while incorporating richer internal signals that capture the evolution of class competition within the network.

4. Experimental Setup

4.1. Model and Datasets

To evaluate error prediction performance, we conduct experiments on multiple image classification benchmarks. Specifically, we used the validation sets of ImageNet-1K [6], CIFAR-100 [13], and Places365 [19], which together provide diversity in dataset size and complexity. For all experiments, we employ a Vision Transformer (ViT-Large) [7] as the base architecture.

4.2. Baselines

We compare against both classical confidence-based methods and recent internal-signal approaches inspired by hallucination detection in large language models. Let $z(x) \in \mathbb{R}^C$ denote the classifier logits over C classes and $p = \mathrm{softmax}(z)$.

4.2.1. Classical Error-Prediction Methods

- **Max logit:** $s_{\text{maxlogit}}(x) = \max_c z_c(x)$.
- **Entropy:** $s_{\text{ent}}(x) = -\sum_{c=1}^{C} p_c(x) \log p_c(x)$.
- **Logit margin:** $s_{\text{margin}}(x) = z_{(1)}(x) - z_{(2)}(x)$, where $z_{(1)} \geq z_{(2)}$ are the top-2 logits.
- **Energy score [17]:**

$$E(x) = -T \log \sum_{c=1}^{C} \exp(z_c(x)/T)\,,$$

  and we use $s_{\text{energy}}(x) = -E(x)$ so that higher scores indicate higher confidence.
- **Mahalanobis [15]:** We follow the Mahalanobis detector of Lee et al., originally proposed for out-of-distribution and adversarial detection. The method models in-distribution training features with a class-conditional Gaussian distribution under a tied-covariance assumption. At inference, it computes layer-wise Mahalanobis scores that measure the proximity of the input feature to the nearest class mean. The resulting layer-wise scores are concatenated and a linear classifier is trained to predict errors.
- **Top-K logits:** To directly compare with our approach, we train a linear classifier on the vector of the top-K logits produced by the base model's final classification head to predict the correctness of its prediction.

4.2.2. LLM-Inspired Internal-Signal Methods

Motivated by recent work on hallucination detection in large language models, we evaluated whether internal activation-based signals transfer to vision error prediction.

- **Linear probing:** We train a linear classifier on intermediate transformer representations to predict whether the model's final prediction is correct. Concretely, we extract a fixed-dimensional hidden representation (e.g., the CLS token at a chosen layer) and fit a linear model for binary error prediction.
- **ACT-ViT [3]:** We adapt ACT-ViT's core idea of *learning over activation tensors*. ACT-ViT constructs an activation tensor by stacking hidden states across layers and tokens, and applies a ViT-inspired architecture over this tensor to predict hallucination/correctness, capturing global patterns across both axes.

4.3. Metrics

Because the error prediction task is highly class-imbalanced, we evaluate performance using the Area Under the Precision–Recall Curve (AUCPR). AUCPR provides a more informative assessment than ROC-based metrics in imbalanced settings by focusing on performance in the minority (error) class.

5. Results

5.1. In-distribution performance

We first evaluate each method under the in-distribution setting, where models are both trained and tested on the same dataset. As shown in Table 1, LogitDynamics achieves the highest AUCPR on two of the three datasets (ImageNet and CIFAR-100) and performs comparably to the best-performing method on Places365.

Overall, logit-based classifiers consistently outperform alternative approaches. In particular, they achieve substantially better results than methods originally designed for hallucination detection in large language models, highlighting the effectiveness of logit-based uncertainty signals for misclassification detection in vision models.

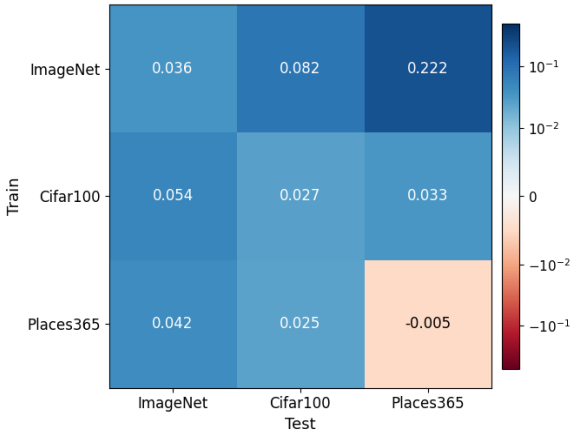
Table 1. In-distribution AUCPR results. Δ denotes LogitDynamics minus the best competing (second-best) method for each dataset.

	ImageNet	CIFAR-100	Places365
Number of classes	1000	100	365
Dataset size	50,000	10,000	36,500
Misclassification rate	20.34%	6.91%	44.5%
Max logit	0.5395	0.3680	0.6969
Entropy	0.5766	0.3967	0.7038
Margin	0.5471	0.4197	0.6728
Energy score	0.4168	0.3580	0.6537
Top-K logits	0.6098	0.4164	**0.7283**
Mahalanobis	0.3244	0.1064	0.4954
ACT-ViT	0.5390	0.1736	0.5719
Linear probing	0.5420	0.3050	0.5736
LogitDynamics	**0.6458**	**0.4430**	0.7232
LogitDynamics Δ	0.0360	0.0266	-0.0051

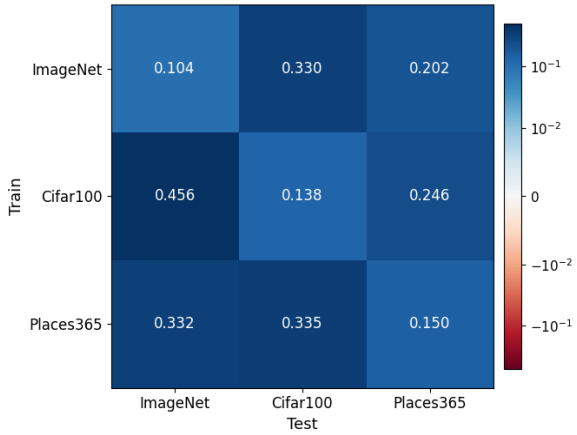
5.2. Cross-dataset performance

We next evaluate cross-dataset generalization using AUCPR differences measured relative to LogitDynamics. For each method, we train the error predictor on one dataset and evaluate it on all three datasets, producing a complete train–test matrix of AUCPR differences relative to LogitDynamics. In Fig. 1, the diagonal entries reflect in-domain gaps, while off-diagonal entries measure how each method's performance changes under distribution shift compared to LogitDynamics, providing a robustness view beyond in-distribution evaluation.

Overall, linear probing exhibits a pronounced decline in cross-dataset performance relative to LogitDynamics.



(a) LogitDynamics - Top-K Logits



(b) LogitDynamics - Linear probing

Figure 1. Cross-dataset AUCPR for each baseline method, reported as the performance difference relative to LogitDynamics (LogitDynamics − method). Rows indicate the training dataset; columns indicate the test dataset.

ACT-ViT demonstrates a similar trend (see Appendix A.1 and Fig. 3). In contrast, logit-based signals show comparatively smaller performance degradations, indicating stronger transferability in vision settings. Among all methods, LogitDynamics consistently exhibits the smallest performance drop under cross-dataset transfer.

5.3. Ablations

We ablate the contribution of the Top-K dynamics statistics (Sec. 3.2) beyond the logit trajectory features (Sec. 3.1). Fig. 2 reports AUCPR(w dynamics) − AUCPR(w/o dynamics), so positive values indicate that adding dynamics features improves AUCPR.

Overall, dynamics features provide little benefit in-distribution but improve cross-dataset transfer: the mean diagonal difference (train=test) is -0.0107, while the mean off-diagonal difference (train≠test) is 0.0155. We hypothesize these features capture stability of class competition under distribution shift, acting as a robustness signal, while adding mild variance in-domain where layer logits are already highly informative.

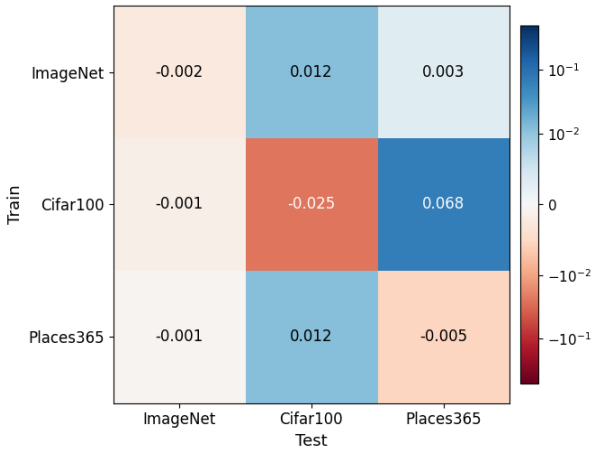


Figure 2. **Ablation study.** Cross-dataset AUCPR differences reported as *with* Top-K dynamics features minus *without* Top-K dynamics features. Positive values indicate that adding Top-K dynamics improves AUCPR, while negative values indicate a decrease.

6. Conclusion

We studied error prediction in Vision Transformers using internal signals available from a single forward pass. Motivated by hallucination-detection methods in large language models, we investigated whether internal signals can improve reliability estimation in vision models. We proposed a simple approach that models the dynamics of class evidence across layers by combining layer-wise logit features with statistics capturing instability among the top-ranked classes.

Experiments across ImageNet-1K, CIFAR-100, and Places365 show that LogitDynamics consistently improves or matches AUCPR over classical logit-based confidence measures and internal-activation baselines, while requiring minimal additional computation and no modification to the backbone network. These results suggest that depth-wise belief dynamics provide a useful and complementary signal for predicting model errors. Future work may explore extending these signals to other architectures, tasks, and distribution shifts.

References

[1] Jonathan Aigrain and Marcin Detyniecki. Detecting adversarial examples and other misclassifications in neural networks by introspection. *arXiv preprint arXiv:1905.09186*, 2019. 1

[2] Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore, 2023. Association for Computational Linguistics. 1

[3] Guy Bar-Shalom, Fabrizio Frasca, Yaniv Galron, Yftah Ziser, and Haggai Maron. Beyond token probes: Hallucination detection via activation tensors with ACT-ViT. In *Advances in Neural Information Processing Systems 38*, 2025. 3

[4] Guy Bar-Shalom, Fabrizio Frasca, Derek Lim, Yoav Gelberg, Yftah Ziser, Ran El-Yaniv, Gal Chechik, and Haggai Maron. Beyond next token probabilities: Learnable, fast detection of hallucinations and data contamination on LLM output distributions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30058–30066, 2026. 1

[5] Charles Corbière, Nicolas Thomé, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems 32*, pages 2902–2913, 2019. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[8] Leo Feng, Mohamed Osama Ahmed, Hossein Hajimirsadeghi, and Amir H. Abdi. Towards better selective classification. In *International Conference on Learning Representations*, 2023. 1

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016. 1

[10] Eduardo Dadalto Câmara Gomes, Marco Romanelli, Georg Pichler, and Pablo Piantanida. A data-driven measure of relative uncertainty for misclassification detection. In *International Conference on Learning Representations*, 2024. 1

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 1

[12] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3301–3310. PMLR, 2019. 2

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 2

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413, 2017. 1

[15] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31*, pages 7167–7177, 2018. 1, 3

[16] Hengyue Liang, Le Peng, and Ju Sun. Selective classification under distribution shifts. *Transactions on Machine Learning Research*, 2024. 1

[17] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems 33*, pages 21464–21475, 2020. 1, 3

[18] Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, 2025. 1

[19] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 2

A. Appendix

A.1. Additional Cross-Dataset Results

ACT-ViT mirrors the behavior of linear probing, demonstrating poor cross-dataset generalization and underperforming LogitDynamics across train–test combinations.

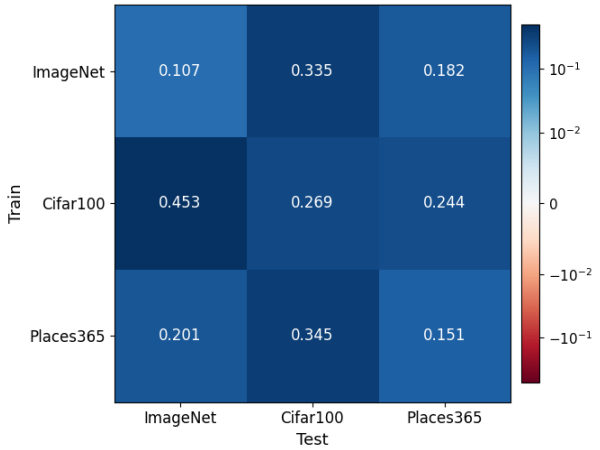


Figure 3. Cross-dataset AUCPR differences relative to LogitDynamics for ACT-ViT. Rows: training dataset; columns: test dataset.

A.2. Top-K Dynamics Feature Definitions

This appendix provides the formal definitions for the Top-K dynamics features described in Sec. 3.2.

Depth-wise logit sequence. We focus on the last L transformer blocks and additionally include the base classifier logits as the terminal element. Let T be the total number of transformer blocks. Define the sequence $\{\tilde{z}_\ell(x)\}_{\ell=1}^{L+1}$ by

$$\tilde{z}_\ell(x) = \begin{cases} z^{\text{head}}_{T-L+\ell}(x), & \ell = 1, \dots, L, \\ z^{\text{clf}}(x), & \ell = L+1, \end{cases}$$

where $z_t^{\text{head}}(x) \in \mathbb{R}^C$ denotes the logits produced by the auxiliary linear head attached to layer t, and $z^{\text{clf}}(x) \in \mathbb{R}^C$ denotes the base model's final classifier logits.

Top-1 and Top-K identities. For each ℓ, define the top-1 identity

$$c_\ell = \arg\max_{c \in \{1,\dots,C\}} \tilde{z}_{\ell,c}(x),$$

and the top-K set

$$S_\ell = \text{TopK}\big(\tilde{z}_\ell(x)\big) \subseteq \{1, \dots, C\},$$

where TopK returns the indices of the K largest components.

Restricted-softmax weights over Top-K. We define normalized weights over the top-K logits by applying a softmax restricted to S_ℓ:

$$w_\ell(i) = \begin{cases} \dfrac{\exp(\tilde{z}_{\ell,i}(x))}{\sum_{j \in S_\ell} \exp(\tilde{z}_{\ell,j}(x))}, & i \in S_\ell, \\ 0, & i \notin S_\ell. \end{cases}$$

In practice, we compute this stably by subtracting $\max_{j \in S_\ell} \tilde{z}_{\ell,j}(x)$ inside the exponent.

Top-1 Switch Rate.

$$\text{SR}(x) = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}[c_\ell \neq c_{\ell+1}].$$

Top-K Weighted Jaccard Similarity. For adjacent depths $(\ell, \ell+1)$ define the weighted intersection mass

$$I_{\ell,\ell+1} = \sum_{i \in S_\ell \cap S_{\ell+1}} \min\{w_\ell(i), w_{\ell+1}(i)\}.$$

The weighted Jaccard similarity is

$$J_w(S_\ell, S_{\ell+1}) = \frac{I_{\ell,\ell+1}}{\sum_{i \in S_\ell} w_\ell(i) + \sum_{i \in S_{\ell+1}} w_{\ell+1}(i) - I_{\ell,\ell+1}}.$$

We report the mean over adjacent pairs:

$$\frac{1}{L} \sum_{\ell=1}^{L} J_w(S_\ell, S_{\ell+1}).$$

Unique Top-K Count.

$$\left| \bigcup_{\ell=1}^{L+1} S_\ell \right|.$$

Top-1 Mode Frequency. Let $n(c) = \sum_{\ell=1}^{L+1} \mathbb{1}[c_\ell = c]$ be the count of each top-1 identity and define $p(c) = n(c)/(L+1)$. The mode frequency is

$$\max_c p(c).$$

Top-1 Entropy. Using the same empirical distribution $p(c)$ over top-1 identities,

$$-\sum_c p(c) \log p(c).$$

Top-1 Unique Count.

$$\left| \{c_\ell\}_{\ell=1}^{L+1} \right|.$$

Top-1 Commitment Depth. Let c_{L+1} be the final element's top-1 identity. Define the earliest depth at which the top-1 remains equal to the final top-1 for all subsequent depths:

$$\ell^\star = \min\{\, \ell \in \{1, \ldots, L+1\} : c_\ell = c_{\ell+1} = \cdots = c_{L+1} \,\}.$$

We normalize the commitment depth as

$$\frac{\ell^\star - 1}{L},$$

which lies in $[0, 1]$ with smaller values indicating earlier commitment.

A.3. Data Splits and Hyperparameters

Splits. For each dataset with N examples, we construct stratified splits with respect to the binary error label $e(x)$ (incorrect vs. correct). We first split the data into an 85% train/validation partition and a 15% held-out test set. The 85% partition is further split into: (i) a *head-training* subset used to train the per-layer linear classification heads, and (ii) a *probe pool* used to train and validate the error predictor. Concretely, we allocate a fraction p_{probe} of the 85% partition to the probe pool (default $p_{\text{probe}} = 0.2$), and the remaining portion to head training. For CIFAR-100, we use a larger probe fraction ($p_{\text{probe}} = 0.3$) due to its smaller sample size and stronger class imbalance in the error labels. Finally, we split the probe pool into 75% probe-train and 25% probe-val. All stages use stratified sampling by $e(x)$.

Auxiliary layer heads. We attach a linear head to each transformer block in a suffix of the network (the last L blocks). Heads are trained with cross-entropy on the head-training split while keeping the ViT backbone frozen. We optimize with AdamW, using a batch size of 512 and weight decay 0.0. In hyperparameter sweeps, we vary the head learning rate in $\{10^{-4}, 2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 7 \cdot 10^{-4}, 10^{-3}\}$ and the number of head-training epochs in $\{2, 5, 7, 10, 12, 16\}$.We select the best configuration based on probe-val AUCPR. For cross-dataset generalization experiments, we extract features using auxiliary heads trained on the *target* dataset (i.e., we train the auxiliary heads on the target dataset's training split with a frozen backbone, and then evaluate the transferred error predictor on target features).

Feature construction. Unless stated otherwise, we use the last $L \in \{1, 3, 5, 7, 9, 12, 16, 20, 24\}$ transformer blocks and include the base classifier logits as the terminal element in the depth sequence. We extract numeric logit features using the predicted-class logit together with the top-K competitor logits, with $K \in \{1, 3, 5, 7, 10\}$; competitors exclude the predicted class for the numeric logit features. Dynamics features (Sec. 3.2) are computed using the raw top-K sets at each depth (without excluding the predicted class). We include the base classifier logits as an additional block of numeric features.

Normalization. All features are standardized using the mean and standard deviation computed on the *probe-train* split only, and we apply the same normalization to probe-val and the held-out test set to avoid leakage. For cross-dataset generalization experiments, we compute normalization statistics using the *target dataset's training split*, and apply this normalization when evaluating on the target dataset.

Error predictor. The error predictor is a linear classifier trained with AdamW on the probe-train split. We train for 100 epochs with learning rate 10^{-3}, batch size 256. Because the error label is imbalanced, we use a positive-class weighting term in the binary cross-entropy loss, where the positive weight is set to $\frac{N_{\text{neg}}}{N_{\text{pos}}}$ computed on the probe-train split.

Model selection. We select hyperparameters using probe-val AUCPR (PR-AUC). Final results are reported on the held-out 15% test set using the same trained heads, normalization statistics, and error predictor.